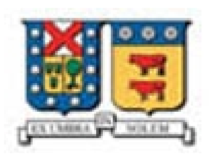


UNIVERSIDAD TECNICA FEDERICO SANTA MARIA
DEPARTAMENTO DE ELECTRONICA
ELO212 Laboratorio de Sistemas Digitales



Laboratorio 09. "Desarrollo de una Interfaz de memoria RAM"

Objetivo Principal:
Aplicar los contenidos vistos sobre máquinas secuenciales y diagramas de temporización para desarrollar una interfaz que controle la memoria RAM de la tarjeta de desarrollo.

Objetivos Específicos:
- Aplicar sentencias específicas del lenguaje ABEL para describir sistemas secuenciales sincrónicos y asincrónicos.
- Estudiar especificaciones de memorias RAM estáticas.
- Conocer y usar las operaciones sobre una memoria estática.
- Separar un bus bidireccional en uno de entrada y otro de salida
- Desarrollar unidad de control para administrar los recursos o camino de datos.
- Desarrollar en forma modular un sistema.
- Administrar la información de control y de datos de un sistema.
- Implementar el diseño en tarjeta de desarrollo.
- Generar vectores de prueba correspondientes.

Preparación previa.

Recursos.

1.- Memorias RAM Estáticas.

En la sección Clases, capítulo 19, se encuentra una detallada descripción de la implementación de una interfaz RAM, que considera la implementación de una máquina de estados para el control de acceso de operaciones de escritura y lectura de la misma. Es imprescindible su lectura y compresión para el desarrollo de las actividades de laboratorio.

2.- Memoria RAM Estática AS7C256.

En la sección Aplicaciones, tarjeta de desarrollo, se encuentran las hojas de características de todos los componentes de la tarjeta de desarrollo. Entre ellas está la correspondiente a la memoria SRAM, que en la tarjeta de desarrollo es el chip Alliance AS7C256-**15JC**. La última sigla es importante ya que permite obtener de la tabla de especificaciones los tiempos de acceso y de ciclo correspondientes.

Especial atención requiere la tabla de verdad presentada en la página 2, que indica el modo de operación de la memoria y la disponibilidad de los datos desde el bus correspondiente.

Los tiempos detallados en las tablas del ciclo de lectura y de escritura también requieren especial atención, ya que permiten ajustar el diseño a las exigencias más restrictivas, que corresponden al los tiempos mínimos (en las tablas se especifican tiempos mínimos y máximos). Además estos tiempos influirán en el período (frecuencia) con que se opere toda la tarjeta de desarrollo, ya que es evidente que se necesita un reloj con un

período de operación menor que los tiempos de acceso y ciclo dados para obtener una eficiencia mayor.

3.- Ajuste de diseños en la CPLD (fit, fitter, fit report).

Al realizar la traducción a lenguaje ABEL de algunos diseños, se pueden presentar problemas de ajuste de los mismos. En este caso, como se quiere implementar una interfaz de memoria, de acuerdo a las especificaciones del fabricante, se deben considerar en el diseño registros de 15 bits para las direcciones, y de 8 bits para los datos, lo que, junto con la lógica de control puede generar una gran cantidad de ecuaciones lógicas, que no puedan ser ajustadas en la CPLD. En este caso conviene aclarar que esto es debido al diseño propiamente tal, por lo que debe modificársele, eliminando variables innecesarias, a fin de obtener el ajuste necesario.

Para ayudar en la tarea de reducir la implementación lógica de un diseño pueden emplearse las directivas *'collapse'* y *'node'* (ver sección Aplicaciones, ABEL reference manual, capítulo 5) que permiten ahorrar recursos de la CLPD.

Se recomienda utilizar diseño jerárquico para el proceso de creación de la interfaz. Es posible visualizar en el **fitter report** los porcentajes de utilización de recursos de la CPLD para un módulo en particular, de manera de tener una idea de cómo se están utilizando los recursos de la tarjeta.

4.- Diseño Propuesto.

En la experiencia 9, año 2002, se propone un diseño de la interfaz a base de una máquina secuencial para sincronizar los eventos de accesos de lectura o escritura, según la operación que se desee. El diagrama de conexiones propuesto en dicho documento contempla la existencia de dos señales para realizar las peticiones de escritura o lectura y una señal de salida que indica cuando un proceso de escritura o lectura de la memoria esté listo (figura 1). En dicha experiencia se propone el diseño de una máquina secuencial que controle los tiempos de acceso de acuerdo al diagrama temporal especificado por el fabricante para implementar la interfaz. En el diagrama que se presenta se agrega con una línea punteada los módulos que deben ser implementados en la CPLD a través de ABEL, con lo que se pueden identificar las entradas y las salidas de la interfaz con respecto a la memoria, que es un chip integrado a la arquitectura de la tarjeta de desarrollo. Se puede observar que hay ciertas señales que constituirán la entrada del módulo, y ciertas señales que serán salida del mismo. Por ejemplo, existe un registro interno del módulo que permite mantener estable la dirección de memoria durante los ciclos del lectura y escritura. Además el registro de escritura (RE) tiene una salida de tercer estado (que se puede implementar con la extensión .OE) para poder usar el bus de datos bidireccional.

En esta oportunidad se propone reemplazar la lógica de la máquina secuencial por la filosofía de diseño utilizada en experiencias anteriores, orientada a concretar un diseño que permita recrear las formas de onda necesarias para cada ciclo, de manera operar

correctamente sobre el bus de datos (leer o escribir) en el instante adecuado, a través del uso de un contador, cuyo reloj debe tener un período tal que asegure los tiempos de acceso para cada proceso, dados por el fabricante.

Debe tomarse en cuenta que el oscilador programable de la tarjeta (pin 9) entrega una frecuencia máxima de 100 [Mhz] con lo que el mínimo período de reloj es de 10 [ns]. De acuerdo a las especificaciones del fabricante, que confiamos haya Ud. leído, el tiempo de ciclo mínimo, tanto de lectura como escritura es de 15 [ns]. Esto significa que cualquier lógica que se implemente para interactuar con la memoria no podrá sacar la máxima eficiencia de la misma, porque el tiempo mínimo que puede pasar para cambiar entre dos estados de cualquier señal es de 10 [ns].

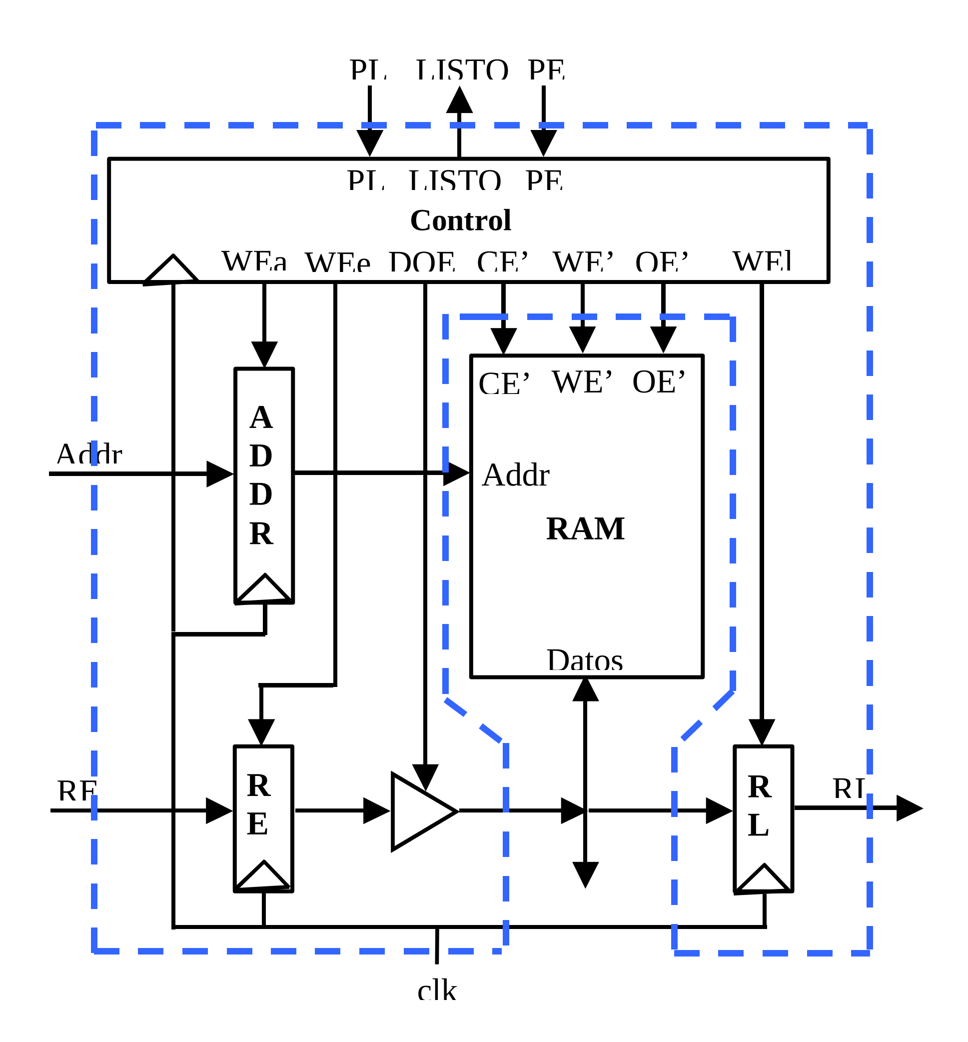


Figura 1. Esquema del diseño propuesto, exp. 9 año 2002. Se observa la línea punteada que encierra los módulos que deben ser implementados en la CPLD.

Actividades Previas.

a) Estudie cuidadosamente las especificaciones del fabricante de la memoria, en cuanto a los tiempos de acceso, ciclo de lectura y escritura, así como los modos de operación de la misma.
b) Estudie las conexiones existentes entre la CPLD y la memoria descritas en el manual de usuario de la tarjeta de desarrollo.
c) Proponga un diseño modular que permita realizar la escritura o lectura confiable de un byte, hacia o desde una dirección específica.
d) Implemente dicho diseño en ABEL.

En el Laboratorio.

a) Pruebe el diseño implementado la parte d) de las actividades previas:

- Grabando y leyendo distintos datos de una misma dirección.
- Grabando toda la memoria con un mismo dato y luego leyéndola.

Puede diseñar una máquina secuencial para realizar estos dos tests, que utilice el diseño implementado en la parte d) de las actividades previas. Para hacer el diseño más simple, se puede utilizar modularidad jerárquica a través de las directivas *interface* y *fuctional_block*.

b) Observe en el osciloscopio mixto las formas de onda generadas por el módulo cuando se está en proceso de lectura y en proceso de escritura.

c) Implemente, en una jerarquía mayor un módulo que capture datos desde el teclado (make codes) y los grabe en una determinada zona de memoria. En modo de escritura, se capturan 10 make codes y se graban en la memoria. En modo de lectura, se controla a través de la puerta paralela cual de los 8 caracteres grabados se visualizan a través del display de 7 segmentos. (Ahorre tiempo y reutilice el decodificador del display y el módulo que captura make codes del teclado, implementados en la experiencia 7.) Observe en osciloscopio las formas de onda generadas (bus de datos y direcciones).